Coupling SPH with a 1-D Boussinesq-type wave model

Christophe Kassiotis*, Martin Ferrand†, Damien Violeau*, Benedict D. Rogers‡,
Peter K. Stansby‡ and Michel Benoit*

*Laboratoire d'Hydraulique Saint-Venant,
Université Paris-Est (Joint Research Unit EDF R&D / CETMEF / École des Ponts ParisTech),
Email: christophe.kassiotis@enpc.fr
†MFEE, EDF R&D, Chatou, France
‡School of MACE, University of Manchester, Manchester,UK

***Abstract*—The high computational cost of SPH remains problematic in dealing with wave propagation, especially when the domains considered are large. In order to overcome this difficulty, we propose to couple 2-D SPH with a 1-D Finite Difference Boussinesq-type model. The latter deals with wave propagations for most of the spatial domain, whereas SPH computations focus on the shoreline or close to off-shore structures, where a complex description of the free-surface is required.**

The re-use of existing codes is achieved using a generic implementation based on Component Technology. The communication between software is ensured by the middleware Component Template Library (CTL) [1], [2]. In order to deal with open domains, open-boundaries have to be implemented for SPH, with water height and velocity varying in space and time. These velocity and water height values are then driven by the Boussinesq-type model.

As an illustration of the one way coupling, we present herein two simple examples of water waves, the first one with a flat bottom, the other one representing a schematic coastal protection.

I. Introduction

The recent events in Japan have dramatically underlined the necessity of accurate predictions for coastal protections. For this kind of civil engineering coastal devices, it is required to compute the flow near the shoreline, where the waves break. But even if the recent trends in the development of GPU-based SPH software seem promising for the computing of large 3D domains, to compute wave propagation on the scale of oceans with SPH is still beyond current computational capabilities. Furthermore, it is often useless, as simplified models are able to represent accurately the wave propagation on most of the domain.

Indeed, the complexity of flows, and especially those at large scales such as wave propagation across ocean make the introduction of simplified models a natural development [3]–[5]. Since the XIXth century, and through the XXth, models such as Saint-Venant [6], Boussinesq [7], [8] and fully Non-Linear Shallow Water equations (NLSW) [9], [10] provide satisfying results in their respective ranges of application (from deep to shallow water).

They are however, by definition, unable to represent accurately the complexity of the flow near the coast, when waves are breaking. The violent hydrodynamics of the wave is often handled by energy dissipation models in the nearshore region (e.g. roller models or sponge layers), often with coefficients that need to be tuned for specific cases. One such example is a study that compares the results from analytical,NLSWE software and two-phase slightly compressible flows solved by VOF strategy for the classical dam break problem [11], and shows the necessity of advanced models for this kind of application.

The development and implementation of appropriate models however is to represent the complex free-surface, evolving in time, with a possible multi-connected domain. Among all the options now available, Smoothed Particle Hydrodynamics (SPH) is offers one of the most attractive approaches. However, 3-D SPH models still often suffer from damping with the waves being dissipated before reaching the coast if no proper treatment is applied.

For these reasons the coupling between any of the wave propagation models and complex free-surface flow strategies seems appropriate to tackle this problem [12], [13]. One of the motivations for this work is to re-use existing codes in order to avoid the long development and validation phase.

The outline of this paper is as follows: in the next section, we present the chosen formulation of the SPH numerical model. In particular, a semi-implicit wall boundary condition is used, as presented in this conference in [14]. In Section III, we detail the Finite Difference Boussinesq-type model used hereafter. The coupling algorithm, the open-boundary conditions required for SPH and the communication between software are detailled in Section IV. In Section V, we present the results of preliminary computations and in Section VI, we finish with some conclusions.

II. SPH Numerical model

A. Continuous equations

We consider a turbulent weakly compressible free-surface flow. The velocity vector, pressure, turbulent kinetic energy

and energy dissipation rate are denoted by $\mathbf{u}$, p, k and ϵ, respectively. Velocities and pressures are Reynolds-averaged, and the effects of turbulent fluctuations are modelled through the concept of eddy viscosity μ_T, estimated from the $k-\epsilon$ model.

The Lagrangian forms of the Reynolds-averaged Navier-Stokes (RANS) and $k-\epsilon$ equations read

$$\begin{aligned}
\frac{d\rho}{dt} &= -\rho \operatorname{div} \mathbf{u} \\
\frac{d\mathbf{u}}{dt} &= -\frac{1}{\rho}\mathbf{grad}\,\tilde{p} + \frac{1}{\rho}\operatorname{div}(\mu_m \mathbf{S}) + \mathbf{g} \\
\frac{d\mathbf{r}}{dt} &= \mathbf{u} \\
\frac{dk}{dt} &= P - \epsilon + \frac{1}{\rho}\operatorname{div}(\mu_k \mathbf{grad}\, k) \\
\frac{d\epsilon}{dt} &= \frac{\epsilon}{k}(C_{\epsilon 1}P - C_{\epsilon 2}\epsilon) + \frac{1}{\rho}\operatorname{div}(\mu_\epsilon \mathbf{grad}\,\epsilon)
\end{aligned} \tag{1}$$

where $\mathbf{g}$ is the gravity acceleration and ρ the fluid density. The modified pressure $\tilde{p}$ and production of turbulent energy P are given by

$$\begin{aligned}
\tilde{p} &= p + \frac{2}{3}\rho k \\
p &= \frac{\rho_0 c_0^2}{\xi}\left[\left(\frac{\rho}{\rho_0}\right)^{\xi} - 1\right] \\
P &= 2\frac{\mu_T}{\rho}\mathbf{S}:\mathbf{S} = \frac{\mu_T}{\rho}\mathbf{S}^2 \\
\mathbf{S} &= \frac{1}{2}\left[\mathbf{grad}\,\mathbf{u} + (\mathbf{grad}\,\mathbf{u})^T\right]
\end{aligned} \tag{2}$$

with c_0 the speed of sound, $\xi = 7$, and $\mathbf{S}$ the rate-of-strain tensor field. Lastly, the dynamic viscosities are given by

$$\begin{aligned}
\mu_m &= \mu + \mu_T \qquad & \mu_T &= C_\mu \rho \frac{k^2}{\epsilon} \\
\mu_k &= \mu + \frac{\mu_T}{\sigma_k} \qquad & \mu_\epsilon &= \mu + \frac{\mu_T}{\sigma_\epsilon}
\end{aligned} \tag{3}$$

where μ is is the fluid dynamic molecular viscosity. The values of the model constants C_μ, $C_{\epsilon 1}$, $C_{\epsilon 2}$, σ_k and σ_ϵ are given in Table I.

C_μ	$C_{\epsilon 1}$	$C_{\epsilon 2}$	σ_k	σ_ϵ
0.09	1.44	1.92	1.0	1.3

Table I
MODEL CONSTANTS

Equations (1) are subject to the following set of boundary conditions at the walls:

$$\begin{aligned}
\left[\frac{\partial}{\partial \mathbf{n}}\left(\frac{p}{\rho} - \mathbf{g}\cdot\mathbf{r}\right)\right]_{\partial\Omega} &= 0 \\
\left(\mu_m \frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right)_{\partial\Omega} &= \mathbf{Q^u} = \boldsymbol{\tau} \\
\left(\mu_k \frac{\partial k}{\partial \mathbf{n}}\right)_{\partial\Omega} &= 0 \\
\left(\mu_\epsilon \frac{\partial \epsilon}{\partial \mathbf{n}}\right)_{\partial\Omega} &= Q^\epsilon
\end{aligned} \tag{4}$$

where $\mathbf{Q^u}$ and Q^ϵ are wall fluxes of momentum and energy dissipation, respectively. The first one is equal to the wall shear stress vector $\boldsymbol{\tau}$ (see [14]). All fluxes are assumed to be zero at the free-surface.

In the next two sections, we will present a renormalized SPH formulation with appropriate boundary fluxes [14].

B. Discrete equations: modified SPH

This modified SPH model is proposed in [14]. The discrete operators are renormalized with a function called $\gamma(\mathbf{r})$, and take account of boundary influence. For this purpose, the discretization of the wall is not based on fictitious particles, but rather on wall segments s and "vertex particles" v (see Figure 3). The corresponding operators are thus marked with a superscript $^\gamma$ and read

$$\begin{aligned}
\mathbf{I}_a^\gamma\{A_b\} &= \frac{1}{\gamma_a}\sum_{b\in F} V_b A_b w_{ab} \\
\mathbf{G}_a^\gamma\{A_{b,s}\} &= \frac{\rho_a}{\gamma_a}\sum_{b\in F} m_b\left(\frac{A_a}{\rho_a^2} + \frac{A_b}{\rho_b^2}\right)\nabla w_{ab} \\
&\quad -\frac{\rho_a}{\gamma_a}\sum_{s\in S}\rho_s\left(\frac{A_a}{\rho_a^2} + \frac{A_s}{\rho_s^2}\right)\nabla\gamma_{as} \\
\widetilde{\mathbf{G}}_a^\gamma\{\mathbf{A}_{b,s}\} &= -\frac{1}{\gamma_a\rho_a}\sum_{b\in F} m_b \mathbf{A}_{ab}\otimes\nabla w_{ab} \\
&\quad +\frac{1}{\gamma_a\rho_a}\sum_{s\in S}\rho_s\mathbf{A}_{as}\otimes\nabla\gamma_{as} \\
\mathbf{L}_a^\gamma\{B_b, A_b, Q_{b,s}^A\} &= \frac{1}{\gamma_a}\sum_{b\in F} V_b(B_a + B_b)\frac{A_{ab}}{r_{ab}^2}\mathbf{r_{ab}}\cdot\nabla w_{ab} \\
&\quad -\frac{1}{\gamma_a}\sum_{s\in S}|\nabla\gamma_{as}|(Q_a^A + Q_s^A)
\end{aligned} \tag{5}$$

(see [14]). As we can see in (6), the discrete operators are now made of two terms each: the first one extends on the set F of fluid and vertex particles, and is simply the corresponding traditional SPH operator and renormalized by a factor γ_a defined later. The second one involves a summation running over the set S of wall segments. For this reason the operators are applied to all data, including particles and segments and denoted by $\{A_{b,s}\}$. The subscripts $_a$ and $_b$ denote quantities relative to a particle, while the $_s$ refers to wall segments.

Note that the discrete Laplacian $\mathbf{L}_a^\gamma$ is now also a function of the fluxes normal to the wall, defined by the following

products

$$\begin{aligned} Q_a^A &= (B\nabla A)_a \cdot \mathbf{n}_s \\ Q_s^A &= (B\nabla A)_s \cdot \mathbf{n}_s \end{aligned} \quad (6)$$

where $\mathbf{n}_s$ is the unit inward normal vector of the segment (see Figure 3). The renormalization function and the contribution of segment s to its gradient are defined by

$$\begin{aligned} \gamma_a &= \int_\Omega w(|\mathbf{r}_a - \mathbf{r}'|)\mathrm{d}^n\mathbf{r}' \\ \nabla\gamma_{as} &= \int_S w(|\mathbf{r}_a - \mathbf{r}_s|)\mathbf{n_s} d^{n-1}\Gamma_S = |\nabla\gamma_{as}|\mathbf{n}_s \end{aligned} \quad (7)$$

where n is the space dimension and $d^{n-1}\Gamma_S$ the surface element of the wall at the point $\mathbf{r}_s$ of the wall. The quantities γ_a are computed from the following governing equation:

$$\frac{d\gamma_a}{dt} = \sum_{s\in S} \nabla\gamma_{ab} \cdot \mathbf{u}_{as} \quad (8)$$

where $\mathbf{u}_{as} = \mathbf{u}_a - \mathbf{u}_s$ is the velocity of the particle a with respect to the segment s. The quantities $\nabla\gamma_{as}$ are computed from exact integrals (see [14] for details).

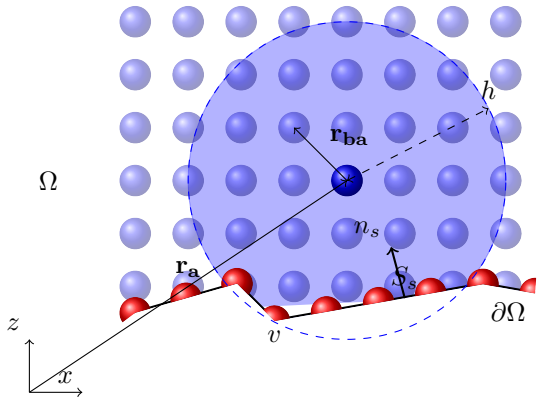


Figure 1. The discretization used in our renormalized SPH model is based on usual particles (a or b), vertex particles v and wall segments s (in two dimensions). The truncation of the kernel is considered through the integral γ (blue area).

As we can see, the present modified SPH model is a bit more complicated than the standard one, but much more efficient for both confined flows [14] and free-surface flows involving inlet/outlet boundaries [15].

III. 1-D Boussinesq-type model

The wave propagation can be described quite accurately using simplified models; for instance, the classical Saint-Venant and Boussinesq models describe the flow with integrated Navier-Stokes equations over the water depth $H = h+\eta$ where h is the water depth at rest, and η the surface elevation (see Figure 2). For the Boussinesq model, only the high order

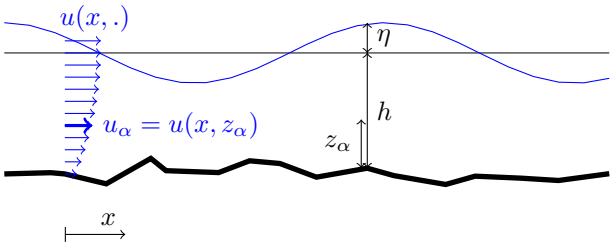


Figure 2. Notations for the 1D vertical Boussinesq-type model.

terms are kept [7]. The mass conservation equation is now written as:

$$\begin{aligned} &\partial_t H + \partial_x H u_\alpha \\ &-\partial_x \left(H \left[\left(\tfrac{\eta^2-\eta h+h^2}{6} - \tfrac{z_\alpha}{2} \right) \partial_x^2 u_\alpha \right.\right. \\ &\left.\left. + \left(\tfrac{\eta-h}{2} - z_\alpha \right) \partial_x \left(\partial_x h u_\alpha + \partial_t h \right) \right] \right) = 0 \end{aligned}$$

and the momentum equation states as:

$$\begin{aligned} &\partial_x u_\alpha + \frac{1}{2}\partial_x u_\alpha^2 + g\partial_x \eta \\ &+\partial_t \left[\frac{z_\alpha}{2}\partial_x^2 u_\alpha + z_\alpha \partial_x \left(\partial_x h u_\alpha + \partial_t h \right) \right. \\ &\left. -\partial_x \left(\frac{\eta}{2}\partial_x^2 u_\alpha + \eta \partial_x \left(\partial_x h u_\alpha + \partial_t h \right) \right) \right] \\ &+\partial_x \left[\partial_t \eta \left(\partial_x h u_\alpha + \partial_t h + \eta \partial_x u_\alpha \right) \right. \\ &+(z_\alpha - \eta) u_\alpha \partial_x \left(\partial_x h u_\alpha + \partial_t h \right) + \frac{z_\alpha^2 - \eta^2}{2} u_\alpha \partial_x^2 u_\alpha \\ &\left. +\frac{1}{2}\left(\partial_x h u_\alpha + \partial_t h + \eta \partial_x u_\alpha \right)^2 \right] = 0 \end{aligned}$$

where u_α is the horizontal velocity at a reference height z_α. We usually consider that $z_\alpha = -0.531h$. As in [16], in order to simplify the notations, we introduce the variable $U = U(u_\alpha) = \partial_t u_\alpha + h\left[b_1 h \partial^3_{x^2 t} u_\alpha + b_2 \partial_x^2 h \partial_t u_\alpha \right]$. Here b_1 and b_2 denote coefficient that depend of the reference height scale (here 0.531). The second hand of the evolution of η and U are noted $E = E(\eta, u_\alpha)$ and $F = F(\eta, u_\alpha)$. They depend of the variables u_α, η as well as their spatial and time derivatives.

The spatial discretization is ensured with a high order Finite Diffence strategy. The time discreatization is based on a 4^{th} order predictor-corrector scheme. In the Algorithm 1, the solving scheme is presented. In this algorithm, we note $X_{i,n}^k$ the value of X discretized at point i, for the (k)-th sub-iteration of time t_n (see [17]).

IV. Coupling algorithm and its implementation

A. Coupling algorithm

In Narayanaswamy *et. al* [13], an explicit staggered coupling algorithm between Boussinesq wave and a SPH model is proposed. In our computations, we also considere the vertical velocity in a buffer zone, and compute in a different way

Algorithm 1 — Boussinesq model implemented in `BSQ_V2P3` [17]

Given: $u_{i,0}$, $\eta_{i,0}$
for $n = 1 \dots N_{\max}$ **do**
  initialize iteration counter $(k) = 0$
  predictor (explicit, order 3):

$$\begin{cases} \eta^{(0)}_{i,n+1} &= \eta_{i,n} + \frac{\Delta t}{12}(23E_{i,n} - 16E_{i,n-1} + 5E_{i,n-2}) \\ U^{(0)}_{i,n+1} &= U_{i,n} + \frac{\Delta t}{12}(23F_{i,n} - 16F_{i,n-1} + 5F_{i,n-2}) \end{cases}$$

  compute velocity $u^{(0)}_{i,n+1}$ from $U^{(0)}_{i,n+1}$
  compute $E^{(0)}_{n+1} = E(\eta^{(0)}_{i,n+1}, u^{(0)}_{i,n+1})$ and $F^{(0)}_{n+1} = F(\eta^{(0)}_{i,n+1}, u^{(0)}_{i,n+1})$
  while $\Delta\eta > 0.0001$ or $\Delta u > 0.0001$ **do**
    corrector (explicit, order 4):

$$\begin{cases} \eta^{(k+1)}_{i,n+1} &= \eta_{i,n} + \frac{\Delta t}{24}(9E^{(k)}_{i,n+1} + 19E_{i,n} - 5E_{i,n-1} \\ & \quad E_{i,n-2}) \\ U^{(k+1)}_{i,n+1} &= U_{i,n} + \frac{\Delta t}{24}(9E^{(k)}_{i,n+1} + 19F_{i,n} - 5F_{i,n-1} \\ & \quad F_{i,n-2}) \end{cases}$$

    compute velocity $u^{(k)}_{i,n+1}$ from $U^{(k)}_{i,n+1}$
    compute $E^{(k)}_{n+1} = E(\eta^{(k)}_{i,n+1}, u^{(k)}_{i,n+1})$ and $F^{(k)}_{n+1} = F(\eta^{(k)}_{i,n+1}, u^{(k)}_{i,n+1})$
    compute error indicator: $\Delta\eta = \frac{\sum_i |\eta^{(k)}_{i,n+1} - \eta^{(0)}_{i,n+1}|}{\sum_i |\eta^{(k)}_{i,n+1}|}$ and $\Delta u = \frac{\sum_i |u^{(k)}_{i,n+1} - u^{(0)}_{i,n+1}|}{\sum_i |u^{(k)}_{i,n+1}|}$
    $(k) \longleftarrow (k+1)$
  end while
end for

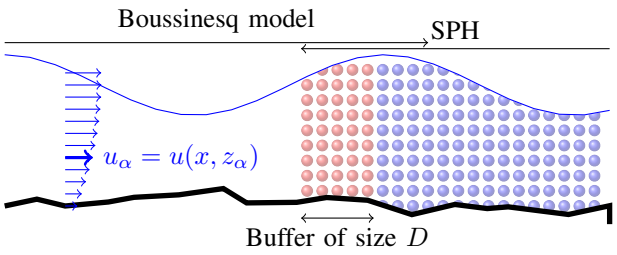


Figure 3. Coupling strategy between Boussinesq model and SPH solver using a buffer zone.

the water heigh in the fluid domain. The Boussinesq solver advances in time with a given time step Δt_{Bsq}. From the Boussinesq solver the velocity profile can be extracted to be imposed at one of the boundary of the SPH solver. For stability reasons, the SPH solver advances in time with smaller time step, denoted Δt_{SPH}. From the SPH solver one obtaines:

- the velocity at the reference depth through an SPH approximation
- the wave height here computed as $2\times$ the mean water particle height.

Algorithm 2 — Explicit coupling between Boussinesq and SPH solvers (2D)

Given initial conditions.
for $n = 1 \dots n_{\max}$ **do**
  $t = n\Delta t_{\text{Bsq}}$
  impose wave height and velocity at reference height on the Boussinesq boundary $(\eta_b, u_{\alpha,b})$
  solve fluid problem with time step Δt_{Bsq}
  get velocity from Boussinesq solver at points (x_i, z_i) on the SPH boundary:

$$\begin{aligned} u(x_i, z_i) &= u_\alpha(x_i) + \partial_x z_\alpha \partial_x h u_\alpha + (z_\alpha - z_i)\partial^2_x h u_\alpha \\ &\quad + z_\alpha \partial_x z_\alpha \partial_x u_\alpha + \tfrac{1}{2}\left(z^2_\alpha - z^2_i\right)\partial^2_x u_\alpha \\ w(x_i, z_i) &= \partial_x u_\alpha(-h - z_i) - u_\alpha \partial_x h \end{aligned}$$

  solve SPH problem from time $t - \frac{\Delta t_{\text{Bsq}}}{2}$ to $t + \frac{\Delta t_{\text{Bsq}}}{2}$
  for $k = 1 \dots k_{\max}$ **do**
    $t = (n - \frac{1}{2})\Delta t_{\text{Bsq}} + k\Delta t_{\text{SPH}}$
    interpolate velocity $(u_i, w_i)^k$ (linear) and impose SPH boundary particle displacement as:

$$\mathbf{r}^{k+1}_i = x^k_i + \Delta t_{\text{SPH}} \mathbf{u}^k_i$$

    solve SPH problem with time step Δt_{SPH}
  end for
  get water height and velocity at the reference height of the Boussinesq boundary.
end for

B. Numerical implementation

A simple way to make software communicate can be to ask one of them to write a file with data and to provide to the another the adapted function to read it. This method suffers of poor CPU time efficiency, and is therefore not easily generalisable. Component-based development requires a middleware layer between clients and services. More precisely, in [18], a Middleware is defined as:

> "providing a standardized, API-like interface that can allow applications on different platforms or written in different languages to interoperate".

Many free and non-free middlewares are currently available on the market; The most well known are CORBA (Common Object Request Broker Architecture), Java™RMI or Microsoft®.NET. However, the field of scientific computing requires high performance in communication, which implies that only a few of the available middleware are of interest for extensive computation. In fact, according to information on performance computing between different types of middleware in [19], only CORBA – among the quoted environments – fulfills the cost requirement, but is known for its complicated syntax.

In the last ten years however, new components like CCA [20], Charm++ [21] or Component Template Library (CTL) where developed specifically for the need of scientific computing and with the aim of simplifying the syntax. In this work, following in the steps of earlier development [1], [2], we will use the CTL as middleware.

Initially part of `ParaFEP` [22], the Component Template Library (CTL) was developed by Dr. R. Niekamp at the Institute für Wissenschaftliches Rechnen (TU-Braunschweig). It is a `C++` template library, like the default `C++` libray – Standard Template Library (STL) – that builds a wrapper or a communication layer around a software, and thus allows so to build components from existing pieces of code. This layer ensures a serialization of the data to be exchanged over a network, and implements, via code generation, an interface defined in a particular header file. Unlike complicated CORBA syntax, the API is here written in C-preprocessor language, that is in a `*.ci` (for Component Interface) file.

The main two advantages of CTL are [23]:

- providing a lightweight that can be used on top of several local (library and thread) or remote communication methods (TCP/IP, MPI and others).
- making the process of writing an application or a service which uses the CTL protocol as transparent as possible. Developers of a service can write its implementation as though they would write a normal local class, with the exception that they need to give the CTL a method to serialize the contained data. Developers of a client only needs then to choose a service within the CTL API (Application Programming Inteface) and how it starts. They can use the objects provided by CTL services as if they were standard local objects.

C. Open-boundary for SPH

Let us now address the question of particle treatment in the buffer area. The usual method to create/delete particles is described in [15]. This method is modified in order to deal with boundaries that can swith between inlet or outlet, and with a varying height. We consider a buffer zone buffer zone on which we impose physical variables. To conserve the same number of particles in buffer zone (defined between X_{min} et $X_{min}+D$), we have to create particles at X_{min} abscissa each times a particle leaves the buffer (reaches the $X_{min}+D$ threshold) to go the fluid domain. For an outlet particle that goes from the fluid to the buffer, the motion is then driven by the imposed values. At the outlet, we delete particles which reach the limit of the fluid domain $x < X_{min}$. The thickness of each buffer zone should be at least the kernel support.

In our case, the following difficulties arise since the buffer zone adresses unsteady B.C.:

1) The velocity is not the same for each particle in the buffer zone, as, according to Boussinesq hypotheses, the horizontal (*resp. vertical*) velocity is a quadratic (*resp. linear*) function of height.
2) The buffer zone can be either an input or an output *during* the simulation.
3) The buffer zone has a varying height.

To update the buffer zone, the following algorithm is applied. We define first the application

$$\mathcal{T} : \mathbf{r}_a \longrightarrow \mathbf{r}_a - D\mathbf{e}_x \tag{9}$$

that translates a given particle from its position $\mathbf{r}_a$ with a vector of the size of the buffer zone. We also not $\mathcal{C}$ the operator that create a new particle.

If F is the set of fluid particles, $F_{\mathcal{B}}$ is the set of fluid particles inside the buffer zone $\mathcal{B}$, and $F_{\overline{\mathcal{B}}}$ the set of fluid particle that are not inside the buffer. At a given time t_{n+1}, the particles in $F^{n+1}_{\mathcal{B}\to\overline{\mathcal{B}}} = F_{\mathcal{B}^n} \cap F_{\overline{\mathcal{B}}^{n+1}}$ are going from the buffer zone to the fluid domain. In the buffer zone, the area where there is a lack of particle can be defined as:

$$\mathcal{B}_\emptyset = \{\mathbf{r} \in F_{\mathcal{B}} | \alpha(\mathbf{r}) < \varepsilon\} \tag{10}$$

where ε is a given (small) parameter and α is a renormalisation parameter. To complete the kernel of fluid particles, the buffer needs to remain full over time, even if particle are going out. Moreover, if the buffer height is varying, we need to destroy or create particles. To create new particle, we use a regularly spaced particles $\hat{F}_{\mathcal{B}}$. To update the buffer zone at t_{n+1}, the following application is used:

$$\mathcal{C}\left(\left[\mathcal{B}_\emptyset \cap \mathcal{T}(F^{n+1}_{\mathcal{B}\to\overline{\mathcal{B}}})\right] \cup \left[\mathcal{B}_\emptyset \cap \hat{F}_{\mathcal{B}}\right]\right) \tag{11}$$

Let us recall that $\mathcal{B}$ is a function of time, as the water height is imposed through a Boussinesq computation.

V. Numerical examples

In this section, the coupling strategy is illustrated with two test cases. The coupling concerns a 2D SPH model and a 1D Boussinesq-type model, using respectively `Spartacus` and `Bsq_V2P3`.

A. Wave propagation over a flat bottom

In this first example, we present our preliminary results of a weak coupling between our two solvers.

The wave propagates over a flat domain. The left part, from $x = -10m$ to $x = 0m$ is computed using the Boussinesq-type model, and the initial water depth is $h = 0.5m$. In the Boussinesq side, waves with a $0.12m$ height and a $T = 0.6s$ period are generated. The spatial discretization is ensured with 250 points, and the time step used for this side as well as for the coupling window is $0.02s$.

For the SPH computation, we consider a speed of sound $c = 20m.s^{-1}$. The buffer zone of the SPH model is of size $x_{\text{buff}} = 6 \times dr$, where $dr = 0.01m$. The SPH domain has a lenght of $2.5m$, and around 12 500 particles are required. The time step may varies over time in order to satisfy SPH stability criterion, and many SPH time steps are required to compute the $0.02s$ of the coupling window.

In Figure 4, we present the results for a weak coupling, where only the Boussinesq side has an influence on the SPH

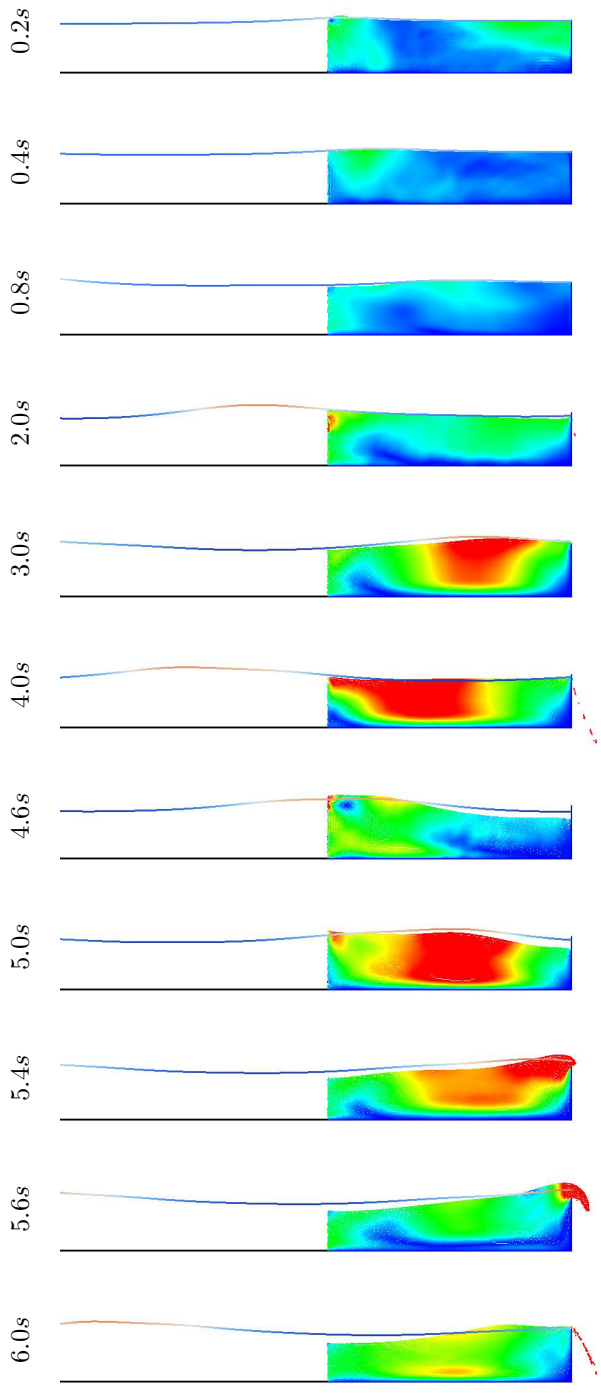


Figure 4. Preliminary results: wave propagation over a flat bottom.

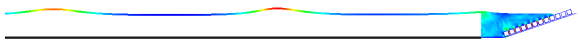
side. We represent the velocities on the SPH side, as well as the water height for the Boussinesq-type model. In order to see the influence of the right boundary on the SPH domain, we also give the water height for a Boussinesq computation over a flat domain without boundary. Let us note that for the first seconds, the weak coupling perform quite well. After a certain time, the wave reflects on the right wall, and it should be required to use a strong coupling strategy so that the SPH influences the Boussinesq computation as well.

B. Wave over a schematic coastal protection

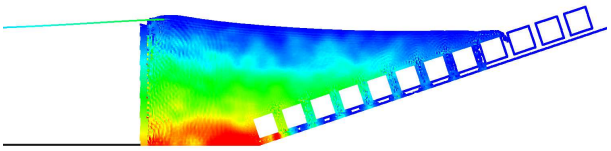
For the second example, the geometry considered is a schematic coastal protection. The water depth is $0.465m$. The domain is flat along $10.5m$. The first $10m$ are modelled with the Boussinesq-type model. Then, a slope $1/3$ starts, covered with squares of size around $0.1m$, spaced from eachother and from the bottom by $0.05m$ (see Figure 5(a)).

In the Boussinesq side, irregular waves are generated. The time step used for the Boussinesq computation and for data exchange between the two models is $0.01s$. For the SPH computation, we consider a speed of sound celerity $c = 20m.s^{-1}$. The buffer zone of the SPH model is of size $x_{\text{buff}} = 6 \times dr$, where $dr = 0.005m$. Around 20 000 particles are required. Note that if the whole domain had been modelled with SPH, around 200 000 should have been required for the same accuracy, and the computational time step would have been multiplied by a factor 1000.

In Figure 5, we present the results for a weak coupling, where only the Boussinesq side has an influence on the SPH side. We represent both the water height and the pressure.



(a) General view of the simulation at $t = 0.2s$. Water height for in the Boussinesq model and velocity field in the SPH zone.



(b) General view of the simulation at $t = 0.2s$. Pressure field for the SPH zone.

Figure 5. Preliminary results: wave propagation and impact on a coastal protection.

VI. Conclusion

We have presented a coupling strategy between an SPH solver and a Finite Difference Boussinesq-type model. From the SPH point of view the coupling requires the implementation of an open boundary with time varying water depths and velocities. Hence, the waves are not built using a classic wave-of, and furthermore, the velocity also has a vertical component in the buffer zone. The implementation, based on the CTL, enables re-use existing codes in a generic way.

In the presented example, the coupling is weak, as only the Boussinesq-type side has an impact on the SPH side. The future work concerns the implementation of a strong coupling, where the Boussinesq model is also influenced by the results of the SPH computations. This strong coupling strategy is obviously more accurate, and removes most of the errors observed at the interface. Another promising possibility of research is to deal with imposing velocities at the Eulerian boundaries with the same kind of strategy used for the solid boundaries based on the renomralization function γ given by (7) in order to avoid the buffer zone.

REFERENCES

[1] C. Kassiotis, R. Niekamp, A. Ibrahimbegović, and H. G. Matthies, "Partitioned procedure for strongly coupled fluid-structure problems. Part II: implementation aspects, re-using existing software as components and nested parallel computations by using CTL," *Computational Mechanics*, vol. Published online, 2010.

[2] R. Niekamp, D. Markovič, A. Ibrahimbegović, H. G. Matthies, and R. L. Taylor, "Multi-scale modelling of heterogeneous structures with inelastic constitutive behavior: Part II–software coupling implementation aspects," *Engineering Computations*, vol. 26, pp. 6–28, 2009.

[3] C. C. Mei, *The Applied Dynamics of Ocean Surface Waves*. Singapore: World Scientific, 1989.

[4] R. J. Sobey, "Linear and Nonlinear Wave Theory," Leichtweiß-Institut für Wasserbau, Braunschweig, Lecture Note, 1998.

[5] J. J. Stoker, *Water Waves: The Mathematical Theory and Applications*. New-York: Wiley-Interscience, 1992.

[6] B. de Saint-Venant, "Théorie du mouvement non-permanent des eaux, avec application aux crues des rivieres et à lintroduction des marées dans leur lit," *Comptes-Rendus de l'Académie des Sciences, Paris, France*, vol. 73, pp. 147–154, 1871.

[7] J. Boussinesq, "Théorie des ondes et des remous qui se propagent le long dun canal rectangulaire horizontal, en communiquant au liquide contenu dans ce canal des vitesses sensiblement pareilles de la surface au fond," *Journal de Mathématiques Pures et Appliquées*, vol. 17, no. 2, pp. 55–108, 1872.

[8] D. H. Peregrine, "Long waves on a beach," *Journal of Fluid Mechanics*, vol. 27, no. 4, pp. 815–827, 1967.

[9] C. Fochesato, S. Grilli, and F. Dias, "Numerical modeling of extreme rogue waves generated by directional energy focusing," *Wave Motion*, vol. 44, pp. 395–416, 2007.

[10] D. Dutykh, "Modélisation mathématique des tsunamis," Thèse de Doctorat, Centre de Mathématiques et Leurs Applications, École Normale Supérieure de Cachan, France, 2008.

[11] D. Dutykh and D. Mitsotakis, "On the relevance of the dam break problem in the context of nonlinear shallow water equations," *Discrete and Continous Dynamical System – Series A*, vol. Accepted, 2009.

[12] C. Lachaume, B. Biausser, S. T. Grilli, P. Fraunie, and S. Guignard, "Modeling of breaking and post-breaking waves on slopes by coupling of bem and vof methods," in *13th Offshore and Polar Engineering Conference (ISOPE)*, 2003, pp. 353–359.

[13] M. S. Narayanaswamy, "A hybrid Boussinesq-SPH wave propagation model with applications to forced waves in rectangular tanks," Ph.D. Thesis, The Johns Hopkins University, 2009.

[14] M. Ferrand, D. Violeau, B. D. Rogers, and D. Laurence, "Consistent wall boundary treatment for laminar and turbulent flows in sph," in *VIth SPHERIC International Workshop*, Hamburg, Germany, June 2011.

[15] O. Mahmood, D. Violeau, M. Ferrand, C. Kassiotis, and C. Denis, "Effect of wall boundary treatment in sph for modelling turbulent flows with inlet/outlet," in *VIth SPHERIC International Workshop*, Hamburg, Germany, June 2011.

[16] G. Wei, J. Kirby, S. Grilli, and R. Subramanya, "A fully nonlinear boussinesq model for surface waves. Part I. Highly nonlinear unsteady waves," *Journal of Fluid Mechanics*, vol. 294, pp. 71–92, 1995.

[17] E. S. Lee, D. Violeau, M. Benoit, R. Issa, D. Laurence, and P. K. Stansby, "Prediction of wave overtopping on coastal structures by using extended Boussinesq and SPH models," *Coastal Engineering*, pp. 4727–4739, 2006.

[18] D. Orenstein, "Quickstudy: Application Programming Interface (API)," *http://www.computerworld.com*, 2000.

[19] R. Niekamp, "Software component architecture," Institut für Wissenschaftliches Rechnen, Germany, Lecture Note, 2005.

[20] J. A. Kohl and D. E. Bernholdt, "CCA home page," 2002, http://www.csm.ornl.gov/cca/.

[21] O. S. Lawlor and G. Zheng, "Charm++ home page," 1999, http://charm.cs.uiuc.edu/research/charm/.

[22] R. Niekamp and E. Stein, "An object-oriented approach for parallel two- and three-dimensional adaptive finite element computations," *Computers and structures*, vol. 80, pp. 317–328, 2002.

[23] B. Bügling, "The component template library protocol and its java implementation," Master Thesis, Technische Universität Braunschweig, Institut for Scientific Computing, 2006, http://www.wire.tu-bs.de/forschung/projekte/ctl/files/ctl_spec.pdf.